PopGen

Hazel Daniels & Nick Carleson

30 March 2021

Following the tutorial found at <https://grunwaldlab.github.io/Population_Genetics_in_R/gbs_analysis.html>

# Reading & manipulating data

NA1.VCF <- read.vcfR("FinalNA.vcf.gz")

## Scanning file to determine attributes.  
## File attributes:  
## meta lines: 66  
## header\_line: 67  
## variant count: 47633  
## column count: 137  
## Meta line 66 read in.  
## All meta lines processed.  
## gt matrix initialized.  
## Character matrix gt created.  
## Character matrix gt rows: 47633  
## Character matrix gt cols: 137  
## skip: 0  
## nrows: 47633  
## row\_num: 0  
## Processed variant 1000Processed variant 2000Processed variant 3000Processed variant 4000Processed variant 5000Processed variant 6000Processed variant 7000Processed variant 8000Processed variant 9000Processed variant 10000Processed variant 11000Processed variant 12000Processed variant 13000Processed variant 14000Processed variant 15000Processed variant 16000Processed variant 17000Processed variant 18000Processed variant 19000Processed variant 20000Processed variant 21000Processed variant 22000Processed variant 23000Processed variant 24000Processed variant 25000Processed variant 26000Processed variant 27000Processed variant 28000Processed variant 29000Processed variant 30000Processed variant 31000Processed variant 32000Processed variant 33000Processed variant 34000Processed variant 35000Processed variant 36000Processed variant 37000Processed variant 38000Processed variant 39000Processed variant 40000Processed variant 41000Processed variant 42000Processed variant 43000Processed variant 44000Processed variant 45000Processed variant 46000Processed variant 47000Processed variant: 47633  
## All variants processed

pop.data <- read\_xlsx("population\_data.wgs.xlsx",1)  
NA1 <- subset(pop.data, pop.data$Lin == "NA1")  
  
#showing which samples need removed  
NA1$ID[!(NA1$ID %in% colnames(NA1.VCF@gt)[-1])]

## [1] "7881-H11" "7612-E11" "7612-D11" "7612-A2" "7612-F6" "7881-A3" "7881-E12"

#removing filtered sites  
NA1 <- filter(NA1, NA1$ID %in% colnames(NA1.VCF@gt)[-1])  
  
#alpha ordering  
NA1 <- arrange(NA1, ID)  
  
#checking order is the same for both  
all(colnames(NA1.VCF@gt)[-1] == NA1$ID)

## [1] TRUE

NA1.gl <- vcfR2genlight(NA1.VCF)  
ploidy(NA1.gl) <- 2  
pop(NA1.gl) <- NA1$Year  
  
NA1.gl

## /// GENLIGHT OBJECT /////////  
##   
## // 128 genotypes, 47,633 binary SNPs, size: 6 Mb  
## 0 (0 %) missing data  
##   
## // Basic content  
## @gen: list of 128 SNPbin  
## @ploidy: ploidy of each individual (range: 2-2)  
##   
## // Optional content  
## @ind.names: 128 individual labels  
## @loc.names: 47633 locus labels  
## @chromosome: factor storing chromosomes of the SNPs  
## @position: integer storing positions of the SNPs  
## @pop: population of each individual (group size range: 12-37)  
## @other: a list containing: elements without names

EU1.VCF <- read.vcfR("FinalEU.vcf.gz")

## Scanning file to determine attributes.  
## File attributes:  
## meta lines: 65  
## header\_line: 66  
## variant count: 51215  
## column count: 139  
## Meta line 65 read in.  
## All meta lines processed.  
## gt matrix initialized.  
## Character matrix gt created.  
## Character matrix gt rows: 51215  
## Character matrix gt cols: 139  
## skip: 0  
## nrows: 51215  
## row\_num: 0  
## Processed variant 1000Processed variant 2000Processed variant 3000Processed variant 4000Processed variant 5000Processed variant 6000Processed variant 7000Processed variant 8000Processed variant 9000Processed variant 10000Processed variant 11000Processed variant 12000Processed variant 13000Processed variant 14000Processed variant 15000Processed variant 16000Processed variant 17000Processed variant 18000Processed variant 19000Processed variant 20000Processed variant 21000Processed variant 22000Processed variant 23000Processed variant 24000Processed variant 25000Processed variant 26000Processed variant 27000Processed variant 28000Processed variant 29000Processed variant 30000Processed variant 31000Processed variant 32000Processed variant 33000Processed variant 34000Processed variant 35000Processed variant 36000Processed variant 37000Processed variant 38000Processed variant 39000Processed variant 40000Processed variant 41000Processed variant 42000Processed variant 43000Processed variant 44000Processed variant 45000Processed variant 46000Processed variant 47000Processed variant 48000Processed variant 49000Processed variant 50000Processed variant 51000Processed variant: 51215  
## All variants processed

EU1 <- subset(pop.data, pop.data$Lin == "EU1")  
  
#showing which samples need removed  
EU1$ID[!(EU1$ID %in% colnames(EU1.VCF@gt)[-1])]

## [1] "7881-B6" "7613-F1" "7881-F10" "7613-D3" "7613-F2" "7613-H7"   
## [7] "7881-G9" "7613-A11" "7613-C3" "7613-F3" "7613-D2" "7881-H10"  
## [13] "7881-E7" "7613-A3" "7881-H6" "7613-H12" "7881-H7" "7613-A8"   
## [19] "7881-H3" "7613-F7" "7881-G4" "7881-C4" "7881-C7" "7613-G3"   
## [25] "7881-C5" "7613-H3" "7881-D6" "7881-G6" "7881-E5" "7613-B3"

#removing filtered sites  
EU1 <- filter(EU1, EU1$ID %in% colnames(EU1.VCF@gt)[-1])  
  
#alpha ordering  
EU1 <- arrange(EU1, ID)  
  
#checking order is the same for both  
all(colnames(EU1.VCF@gt)[-1] == EU1$ID)

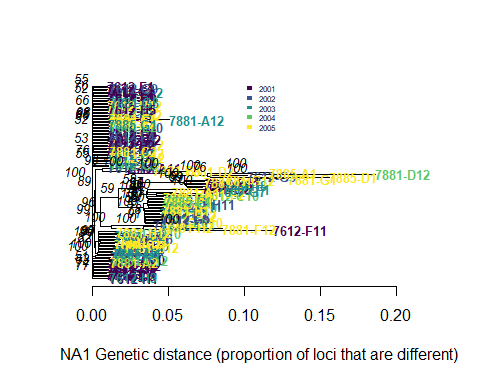
## [1] TRUE

EU1.gl <- vcfR2genlight(EU1.VCF)  
ploidy(EU1.gl) <- 2  
pop(EU1.gl) <- EU1$Year  
  
EU1.gl

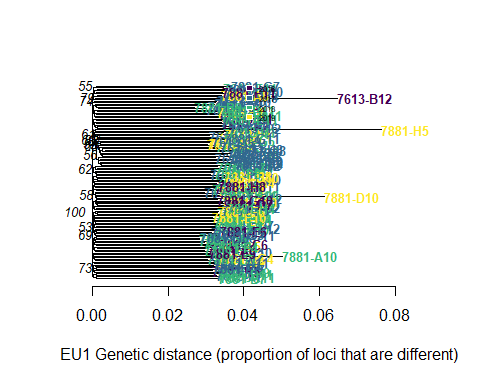
## /// GENLIGHT OBJECT /////////  
##   
## // 130 genotypes, 51,215 binary SNPs, size: 6.4 Mb  
## 0 (0 %) missing data  
##   
## // Basic content  
## @gen: list of 130 SNPbin  
## @ploidy: ploidy of each individual (range: 2-2)  
##   
## // Optional content  
## @ind.names: 130 individual labels  
## @loc.names: 51215 locus labels  
## @chromosome: factor storing chromosomes of the SNPs  
## @position: integer storing positions of the SNPs  
## @pop: population of each individual (group size range: 15-58)  
## @other: a list containing: elements without names

# Distance Tree

treeNA1 <- aboot(NA1.gl, tree = "nj",   
 distance = bitwise.dist,   
 sample = 100, showtree = F,   
 cutoff = 50, quiet = T)  
  
colsNA <- viridis(n = nPop(NA1.gl), begin = 0, end = 1 )  
  
#setting up tree  
plot.phylo(treeNA1, cex = 0.8, font = 2, adj = 0,   
 tip.color = colsNA[pop(NA1.gl)])  
#add node labels (left)  
nodelabels(treeNA1$node.label, adj = c(1.3, -0.5),   
 frame = "n", cex = 0.8,font = 3, xpd = TRUE)  
#add legend (right)  
legend('top', legend = c("2001","2002","2003","2004","2005"),  
 fill = colsNA, border = FALSE, bty = "n", cex = 0.5)  
#add bottom axis  
axis(side = 1)  
#add title  
title(xlab = "NA1 Genetic distance (proportion of loci that are different)")

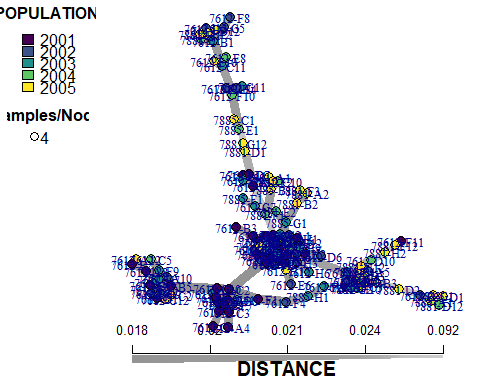


treeEU1 <- aboot(EU1.gl, tree = "nj",   
 distance = bitwise.dist,   
 sample = 100, showtree = F,   
 cutoff = 50, quiet = T)  
  
colsEU <- viridis(n = nPop(EU1.gl), begin = 0, end = 1 )  
  
#setting up tree  
plot.phylo(treeEU1, cex = 0.8, font = 2, adj = 0,   
 tip.color = colsEU[pop(EU1.gl)])  
#add node labels (left)  
nodelabels(treeEU1$node.label, adj = c(1.3, -0.5),   
 frame = "n", cex = 0.8,font = 3, xpd = TRUE)  
#add legend (right)  
legend('top', legend = c("2016","2017","2018","2019"),  
 fill = colsEU, border = FALSE, bty = "n", cex = 0.5)  
#add bottom axis  
axis(side = 1)  
#add title  
title(xlab = "EU1 Genetic distance (proportion of loci that are different)")

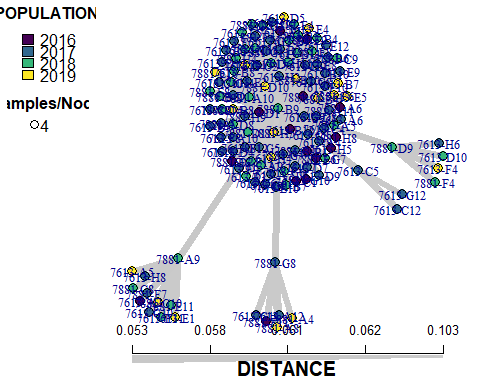


# Minimum Spanning Network

NA1.dist <- bitwise.dist(NA1.gl)  
NA1.msn <- poppr.msn(NA1.gl, NA1.dist, showplot = FALSE, include.ties = T)  
  
node.NA1 <- rep(2, times = nInd(NA1.gl))  
names(node.NA1) <- indNames(NA1.gl)  
vertex.attributes(NA1.msn$graph)$size <- node.NA1  
  
set.seed(47)  
plot\_poppr\_msn(NA1.gl, NA1.msn , gadj = 70,  
 palette = viridis(n = nPop(NA1.gl),   
 begin = 0, end = 1))

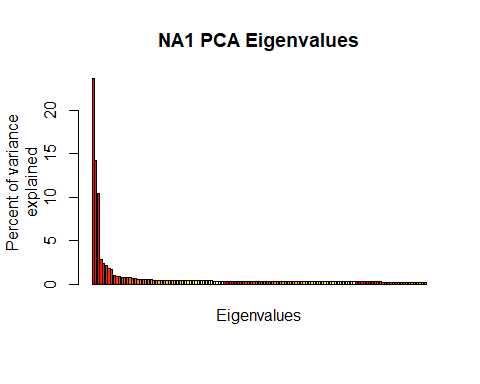


EU1.dist <- bitwise.dist(EU1.gl)  
EU1.msn <- poppr.msn(EU1.gl, EU1.dist, showplot = FALSE, include.ties = T)  
  
node.EU1 <- rep(2, times = nInd(EU1.gl))  
names(node.EU1) <- indNames(EU1.gl)  
vertex.attributes(EU1.msn$graph)$size <- node.EU1  
  
set.seed(47)  
plot\_poppr\_msn(EU1.gl, EU1.msn , gadj = 70,  
 palette = viridis(n = nPop(EU1.gl),   
 begin = 0, end = 1))



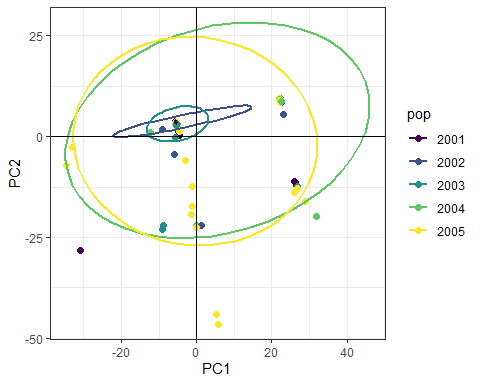
# Principal Components

NA1.pca <- glPca(NA1.gl, nf = 3)  
barplot(100\*NA1.pca$eig/sum(NA1.pca$eig),   
 col = heat.colors(50),   
 main="NA1 PCA Eigenvalues")  
title(ylab="Percent of variance\nexplained", line = 2)  
title(xlab="Eigenvalues", line = 1)

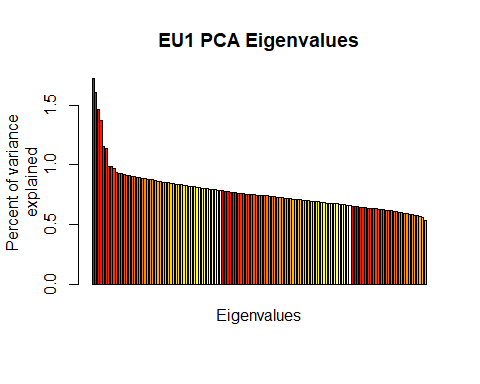


NA1.pca.scores <- as.data.frame(NA1.pca$scores)  
NA1.pca.scores$pop <- pop(NA1.gl)  
  
set.seed(47)  
pp <- ggplot(NA1.pca.scores, aes(x=PC1, y=PC2, colour=pop))+  
 geom\_point(size=2)+   
 stat\_ellipse(level = 0.95, size = 1)+  
 scale\_color\_manual(values = colsNA)+  
 geom\_hline(yintercept = 0)+  
 geom\_vline(xintercept = 0)+   
 theme\_bw()  
pp

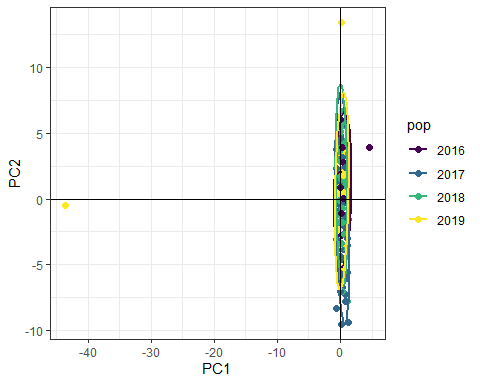
## Warning in MASS::cov.trob(data[, vars]): Probable convergence failure  
  
## Warning in MASS::cov.trob(data[, vars]): Probable convergence failure



EU1.pca <- glPca(EU1.gl, nf = 3)  
barplot(100\*EU1.pca$eig/sum(EU1.pca$eig),   
 col = heat.colors(50),   
 main="EU1 PCA Eigenvalues")  
title(ylab="Percent of variance\nexplained", line = 2)  
title(xlab="Eigenvalues", line = 1)

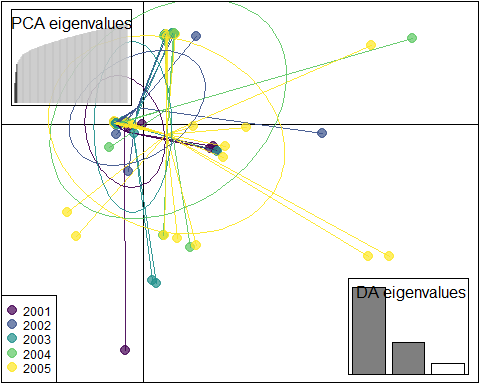


EU1.pca.scores <- as.data.frame(EU1.pca$scores)  
EU1.pca.scores$pop <- pop(EU1.gl)  
  
set.seed(47)  
qq <- ggplot(EU1.pca.scores, aes(x=PC1, y=PC2, colour=pop))+  
 geom\_point(size=2)+   
 stat\_ellipse(level = 0.95, size = 1)+  
 scale\_color\_manual(values = colsEU)+  
 geom\_hline(yintercept = 0)+  
 geom\_vline(xintercept = 0)+   
 theme\_bw()  
qq

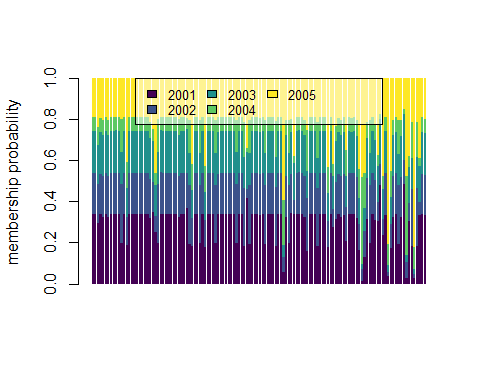


# DAPC

NA1.dapc <- dapc(NA1.gl, n.pca = 3, n.da = 2)  
  
scatter(NA1.dapc, col = colsNA, cex = 2, legend = TRUE, clabel = F, posi.leg = "bottomleft", scree.pca = TRUE,  
 posi.pca = "topleft", cleg = 0.75)



compoplot(NA1.dapc,col = colsNA, posi = 'top')



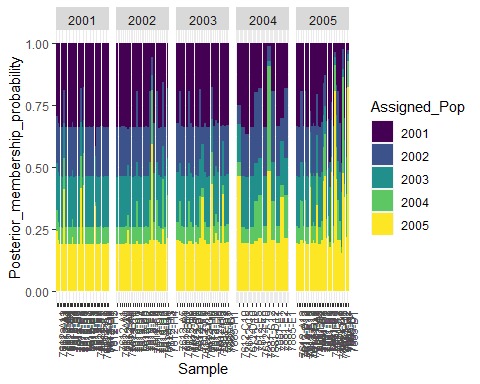
dapc.results.na <- as.data.frame(NA1.dapc$posterior)  
dapc.results.na$pop <- pop(NA1.gl)  
dapc.results.na$indNames <- rownames(dapc.results.na)  
dapc.results.na <- pivot\_longer(dapc.results.na, -c(pop, indNames))  
  
head(dapc.results.na, n = 6)

## # A tibble: 6 x 4  
## pop indNames name value  
## <fct> <chr> <chr> <dbl>  
## 1 2002 7612-A1 2001 0.339   
## 2 2002 7612-A1 2002 0.200   
## 3 2002 7612-A1 2003 0.205   
## 4 2002 7612-A1 2004 0.0676  
## 5 2002 7612-A1 2005 0.187   
## 6 2002 7612-A10 2001 0.338

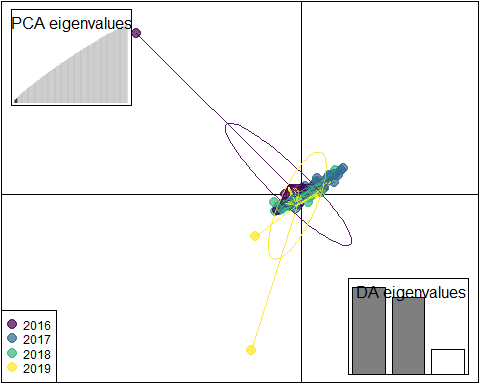
colnames(dapc.results.na) <- c("Original\_Pop","Sample","Assigned\_Pop","Posterior\_membership\_probability")  
  
head(dapc.results.na, n = 6)

## # A tibble: 6 x 4  
## Original\_Pop Sample Assigned\_Pop Posterior\_membership\_probability  
## <fct> <chr> <chr> <dbl>  
## 1 2002 7612-A1 2001 0.339   
## 2 2002 7612-A1 2002 0.200   
## 3 2002 7612-A1 2003 0.205   
## 4 2002 7612-A1 2004 0.0676  
## 5 2002 7612-A1 2005 0.187   
## 6 2002 7612-A10 2001 0.338

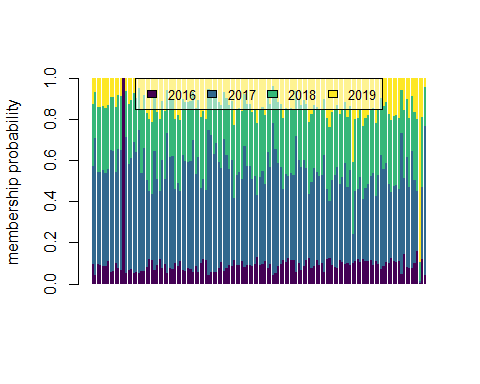
rr <- ggplot(dapc.results.na, aes(x=Sample, y=Posterior\_membership\_probability, fill=Assigned\_Pop))+   
 geom\_bar(stat='identity')+   
 scale\_fill\_manual(values = colsNA)+  
 facet\_grid(~Original\_Pop, scales = "free")+  
 theme(axis.text.x = element\_text(angle = 90,   
 hjust = 1,   
 size = 8))  
rr



EU1.dapc <- dapc(EU1.gl, n.pca = 3, n.da = 2)  
  
scatter(EU1.dapc, col = colsEU, cex = 2, legend = TRUE, clabel = F, posi.leg = "bottomleft", scree.pca = TRUE,  
 posi.pca = "topleft", cleg = 0.75)



compoplot(EU1.dapc,col = colsEU, posi = 'top')



dapc.results.eu <- as.data.frame(EU1.dapc$posterior)  
dapc.results.eu$pop <- pop(EU1.gl)  
dapc.results.eu$indNames <- rownames(dapc.results.eu)  
dapc.results.eu <- pivot\_longer(dapc.results.eu, -c(pop, indNames))  
  
head(dapc.results.eu, n = 6)

## # A tibble: 6 x 4  
## pop indNames name value  
## <fct> <chr> <chr> <dbl>  
## 1 2016 7613-A1 2016 0.0947  
## 2 2016 7613-A1 2017 0.477   
## 3 2016 7613-A1 2018 0.304   
## 4 2016 7613-A1 2019 0.124   
## 5 2017 7613-A10 2016 0.0425  
## 6 2017 7613-A10 2017 0.667

colnames(dapc.results.eu) <- c("Original\_Pop","Sample","Assigned\_Pop","Posterior\_membership\_probability")  
  
head(dapc.results.eu, n = 6)

## # A tibble: 6 x 4  
## Original\_Pop Sample Assigned\_Pop Posterior\_membership\_probability  
## <fct> <chr> <chr> <dbl>  
## 1 2016 7613-A1 2016 0.0947  
## 2 2016 7613-A1 2017 0.477   
## 3 2016 7613-A1 2018 0.304   
## 4 2016 7613-A1 2019 0.124   
## 5 2017 7613-A10 2016 0.0425  
## 6 2017 7613-A10 2017 0.667

ss <- ggplot(dapc.results.eu, aes(x=Sample, y=Posterior\_membership\_probability, fill=Assigned\_Pop))+   
 geom\_bar(stat='identity')+   
 scale\_fill\_manual(values = colsEU)+  
 facet\_grid(~Original\_Pop, scales = "free")+  
 theme(axis.text.x = element\_text(angle = 90,   
 hjust = 1,   
 size = 8))  
ss

